FACE DETECTION AND RECOGNITION USING SKIN SEGMENTATION AND ELASTIC BUNCH GRAPH MATCHING

A Thesis submitted in partial fulfillment of the

requirements for the degree

of

BACHELOR OF ELECTRICAL ENGINEERING

By

SAYANTAN SARKAR

Roll No. : 107EE005

SUBHRADEEP KAYAL

Roll No. : 107EE033

Guided by

PROF. DIPTI PATRA

DEPARTMENT OF ELECTRICAL ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA, ODISHA

MAY, 2011

Dedicated to

the Image Processing Community

National Institute of Technology, Rourkela

Rourkela-769008, Odisha

CERTIFICATE

This is to certify that the thesis entitled **"Face Detection and Recognition using Skin Segmentation and Elastic Bunch Graph Matching"** is submitted by **Mr. Sayantan Sarkar (107EE005)** and **Mr. Subhradeep Kayal (107EE033)** to the Department of Electrical Engineering, National Institute of Technology, Rourkela, for the award of the B.Tech degree in Electrical Engineering, is a bonafide record of work carried out by them under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any institute or university for the award of any degree or diploma to the best of my knowledge.

PROF. DIPTI PATRA
PROJECT GUIDE
DEPT. OF ELECTRICAL ENGG.
NIT, ROURKELA

PROF. B. D. SUBUDHI
HEAD OF DEPT.
DEPT. OF ELECTRICAL ENGG.
NIT, ROURKELA

ACKNOWLEDGEMENT

We would like to express our appreciation towards all the people who have contributed their precious time and effort to help us in our endeavour. Without them it would not have been possible to understand and complete the project.

We would also like to thank **Prof. Dipti Patra**, Department of Electrical Engineering, our project supervisor for her support, motivation and encouragement throughout the period this work was carried out. Her readiness for consultation, educative comments, concern and assistance have been invaluable.

We are grateful to **Dr. B. D. Subudhi**, Head of the Department, Electrical Engineering, for providing us with the necessary facilities as and when required.

Place: Rourkela — Sayantan Sarkar (107EE005)

Date: 9th May, 2011 — Subhradeep Kayal (107EE033)

ABSTRACT

Recently, face detection and recognition is attracting a lot of interest in areas such as network security, content indexing and retrieval, and video compression, because 'people' are the object of attention in a lot of video or images. To perform such real-time detection and recognition, novel algorithms are needed, which better current efficiencies and speeds. This project is aimed at developing an efficient algorithm for face detection and recognition.

This project is divided into two parts, the detection of a face from a complex environment and the subsequent recognition by comparison. For the detection portion, we present an algorithm based on skin segmentation, morphological operators and template matching. The skin segmentation isolates the face-like regions in a complex image and the following operations of morphology and template matching help reject false matches and extract faces from regions containing multiple faces.

For the recognition of the face, we have chosen to use the 'EGBM' (Elastic Bunch Graph Matching) algorithm. For identifying faces, this system uses single images out of a database having one image per person. The task is complex because of variation in terms of position, size, expression, and pose. The system decreases this variance by extracting face descriptions in the form of image graphs. In this, the node points (chosen as eyes, nose, lips and chin) are described by sets of wavelet components (called 'jets'). Image graph extraction is based on an approach called the 'bunch graph', which is constructed from a set of sample image graphs. Recognition is based on a directly comparing these graphs. The advantage of this method is in its tolerance to lighting conditions and requirement of less number of images per person in the database for comparison.

Contents

List of Figures and Equations 8
Chapter – I 11
Introduction and History 11
1.1 The Development of Face Recognition 12
1.2 General Concepts and Approaches 13
1.3 Popular Algorithms 14
1.3.1 PCA or Principal Components Analysis 14
1.3.2 LDA or Linear Discriminant Analysis 15
1.3.3 SVM or Support Vector Machine 15
1.3.4 EBGM or Elastic Bunch Graph Matching 16
1.3.5 Trace Transform (Petrou & Kadyrov) 17
Chapter – II 18
Fundamentals 18
2.1 The digital image and its neighbourhood operations 18
2.1.1 Operations 20
2.1.2 Connectivity and CCA 21
2.2 Image Enhancement Tool: Contrast Stretching 22
2.3 Mathematical Tools 24
2.3.1 Convolution 24
2.3.2 Mathematical Morphology (Gonzalez & Woods) 25
2.3.2.1 Dilation 25
2.3.2.2 Erosion 26
2.3.2.3 Opening 27
2.3.2.4 Hole filling 29
2.3.3 Image Transforms: Gabor Filter and Gabor Transform 30
2.3.4 Euler number (Saveliev) 32
2.4 The HSV Colour Space 32
Chapter – III 34
Detection 34
3.1 Skin Segmentation 34

3.2 Removal of unwanted regions and noise 38
3.2.1 Morphological Cleaning 38
3.2.2 Connected Regions Analysis 42
3.2.2.1 Rejections based on Geometry 42
3.2.2.2 Euler Characteristic based Rejection 44
3.3 Template Matching 45
3.4 Results of the Detection Process 47
Chapter – IV 49
Recognition 49
4.1 Elastic Bunch Graph Matching or EBGM 49
4.1.1 Correlation between image and graph 51
4.1.2 Preprocessing (Gabor Wavelets) 52
4.1.2.1 Comparison between Jets 57
4.1.2.2 Calculation of Displacement, d 61
4.1.2.3 Face graph Representation 63
4.1.3 Face Bunch Graphing 64
4.1.4 Elastic Bunch Graph Matching 66
4.1.4.1 Manual generation of graph 67
4.1.4.2 Matching using Graph Similarity Function 69
4.1.5 Face Recognition based on Similarity Measure 70
4.1.6 Result analysis for Facial Recognition 73
4.1.7 Matching Accuracy 76
4.1.8 Limitations 77
4.2 Databases Used 77
Chapter – V 79
Conclusion 79
Chapter – VI 80

List of Figures and Equations

Figure 1 Processes in face recognition 13
Figure 2 Feature Vectors Derived using 'eigenfaces' 14
Figure 3 Transforming into a graph in EBGM 16
Figure 4 Characteristics of line, p, θ 17
Figure 5 A digital image and its parameters 19
Figure 6 4-neighbourhood and 8-neighbourhood 20
Figure 7 Example of a 4-connected region 21
Figure 8 Portion of an image after CCA. The boxes show all the connected areas which are represented by the pixel ID's as shown 22
Figure 9 Functions used in contrast stretching, clockwise from top: Sigmoid function, Hard limiter, Piecewise linear 23
Figure 10 Result of adaptive contrast stretching: original (left) and enhanced (right) 24
Figure 11 Shows the effect of 'dilation' 26
Figure 12 Shows the effect of 'erosion' 27
Figure 13 An example of 'opening' 28
Figure 14 The process of 'hole-filling' 30
Figure 15 Gabor filters aligned at *0* and *45* degrees respectively, used for filtering oriented edges 31
Figure 16 Average histogram for the 'hue' component 36
Figure 17 Histogram for the 'saturation' component as obtained from the training face images 36
Figure 18 Original image 37
Figure 19 Image in HSV space 37
Figure 20 After histogram based skin segmentation 38
Figure 21 Sequence of steps to 'clean' the image 39
Figure 22 Disk shaped structuring element of unity radius 40
Figure 23 Image after morphological operations have been performed 41
Figure 24 Mask image applied to original 41
Figure 25 Sequence of steps for 'rejection based on geometry' 43
Figure 26 Image contains less noise after Euler number based filtering 45
Figure 27 (top) Training images and (bottom) Average image template 46
Figure 28 Algorithm for template matching 47
Figure 29 Detection process block diagram 47
Figure 30 Original 48
Figure 31 Detected faces 48

Figure 32 Gabor kernels for orientations $\mu = 0$ and 1 at frequency $v = 0$ 54
Figure 33 Gabor kernels for orientations $\mu = 2$ and 3 at frequency $v = 0$ 54
Figure 34 Gabor kernels for orientations $\mu = 4$ and 5 at frequency $v = 0$ 55
Figure 35 Gabor kernels for orientations $\mu = 6$ and 7 at frequency $v = 0$ 55
Figure 36 Gabor kernels for orientations $\mu = 0$ and at frequency $v = 1$ and 2 55
Figure 37 Gabor kernels for orientations $\mu = 0$ and at frequency $v = 3$ and 4 56
Figure 38 Original image (right) and Gabor wavelet transformed image (left) for person 1 in database 57
Figure 39 Original image (right) and Gabor wavelet transformed image (left) for person 2 in database 58
Figure 40 Original image (right) and Gabor wavelet transformed image (left) for person 3 in database 58
Figure 41 Original image (right) and Gabor wavelet transformed image (left) for person 4 in database 59
Figure 42 Original image (right) and Gabor wavelet transformed image (left) for person 5 in database 59
Figure 43 The face graph that can be generated with the considered set of fiducial points 64
Figure 44 Face bunch graph generated using a set of model face image graph to represent an individual person..... 66
Figure 45 Recognition network 72
Figure 46 Input model image of Person 1 (left) and Person 2 (right) 73
Figure 47 Input model image of Person 3 (left) and Person 4 (right) 73
Figure 48 Input model image of Person 5 74
Figure 49 Person 2 face bunch graph and Person 4 input image graph have similarity measure of 0.9985<T2 75
Figure 50 Person 3 face bunch graph and Person 4 input image graph has similarity measure of 0.9968<T3 75
Figure 51 Person 3 face bunch graph and Person 4 input image graph has similarity measure of 0.9968<T3 76
Figure 52 Person 4 face bunch graph and Person 3 input image graph has similarity measure of 0.9991> T4, hence causes over identification of Person 3 as Person 3 and Person 4. 77

Equation 1 Convolution of two functions *a* and *b* 24
Equation 2 Convolution in continuous 2D space 25

Equation 3 Convolution in discrete 2D space 25
Equation 4 Dilation 25
Equation 5 Erosion 26
Equation 6 Opening 27
Equation 7 Hole filling 29
Equation 8 Gabor Filter in 2D 30
Equation 9 Rotated parameters in Gabor filter 31
Equation 10 The Gabor transform 31
Equation 11 Finding out the Euler number 32
Equation 12 Conversion to HSV from RGB 33
Equation 13 Convolution with Gabor Kernels to generate wavelet transformed image 53
Equation 14 Family of Gabor Kernels for *j* varying from 0 to 39 53
Equation 15 Wave vector *kj* for *j* varying from 1 to 39 53
Equation 16 Similarity measure 60
Equation 17 Similarity Function *S* for jets including phase 61
Equation 18 Taylor Series expansion of phase included similarity function 62
Equation 19 Displacement Vector 62
Equation 20 Graph Similarity Measure between image graph and bunch graph 69
Equation 21 Recognition Index for the database used for analysis 71

Chapter – I

Introduction and History

Humans inherently use faces to recognize individuals and now, advancements in computing capabilities over the past few decades enable similar recognitions automatically. Face recognition algorithms have, over the years, developed from simple geometric models to complex mathematical representations and sophisticated vector matching processes. Technical advancements in computing and image acquisition techniques have propelled face recognition technology into the spotlight and researchers are interested more than ever to develop novel techniques to efficiently and robustly recognize faces to give finally help in providing computers with cognitive capabilities.

Face detection and recognition has, thus, developed into a very active research area specializing on how to extract and recognize faces within images or videos. 'Face detection' is the process of finding and extracting facial features within images or videos. 'Face recognition' is the process of matching the detected facial features to one of many the faces known to the file system. With face recognition and detection systems finding use in surveillance, automatic photography and tracking, novel and robust algorithms improving detection rate and recognition are coming up.

1.1 The Development of Face Recognition

Automatic recognition of human faces is a fairly new concept. First instances of a semi-automated system for detecting faces came in the 1960's which required a 'supervisor' or an administrator to locate features such as the eyes, the nose, the mouth, the ears, etc, on the photographs before it could calculate distances to some reference point and compare. In the 1970's Goldstein, Harmon and Lesk (Goldstein, Harmon, & Lesk, May 1971) used markers such as hair colour and lip thickness to automate the recognition process. Still, the measurements and locations were manually calculated. In 1988, Kirby and Sirovich applied 'principal component analysis', which was a standard algebraic technique, to the problem of automatic face recognition. This was a breakthrough as it showed that less than 100 values were required to correctly code an aligned normalized image (Sirovich & Kirby, 1987). In 1991, Turk and Pentland showed that while using the 'eigenfaces method', the error could be used to find out the faces in an image (Turk & Pentland, 1991). This was a significant discovery and it enabled real-time automated face recognition systems. Even though the approach was restricted by lighting and other environmental noise factors, it contributed largely in creating interest in furthering development of automatic face identification techniques.

Today, face recognition is actively being used to minimize passport fraud, support law enforcement agencies, identify missing children and combat identity theft. The applications of face recognition are boundless.

1.2 General Concepts and Approaches

The problem of face recognition can be dealt with by including –

→ A 'face detector', which finds the exact locations of human faces from a normal picture in a simple or a complex background.

→ A 'face identifier' that determines who the person is.

Typically, the algorithms for face detection and recognition fall under any one of the following broad categories (Xiong) –

1. Knowledge-based methods: encode what makes a typical face, e.g., the association between facial features.
2. Feature-invariant approaches: aim to find structure features of a face that do not change even when pose, viewpoint or lighting conditions vary.
3. Template matching: comparison with several stored standard patterns to describe the face as a whole or the facial features separately.
4. Appearance-based methods: the models are learned from a set of training images that capture the representative variability of faces.

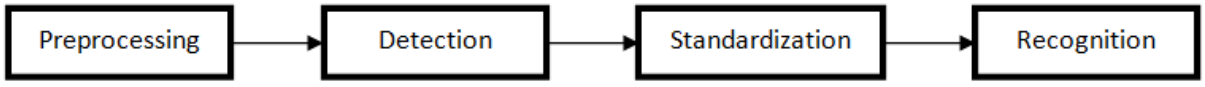

Figure 1 Processes in face recognition

1.3 Popular Algorithms

1.3.1 PCA or Principal Components Analysis

PCA, normally referred to as the use of ‘eigenfaces’, reduces the dimension of data by data compression and exposes the effective low dimensional structure of facial patterns. The images must be of same size and must also be normalized to line up the eyes and mouth of subjects within the images. The resolution of the images into what are called ‘eigenfaces’ (which are analogous to ‘eigenvectors’ in a matrix), which are orthogonal unrelated components, removes all useless extra information. Each face image may be represented as a weighted sum (‘feature vectors’) of eigenfaces, which are stored in a one-dimensional array. Then the distance between the feature vectors of the test image and the image in database is calculated, and match is found based on a minimum distance criterion.

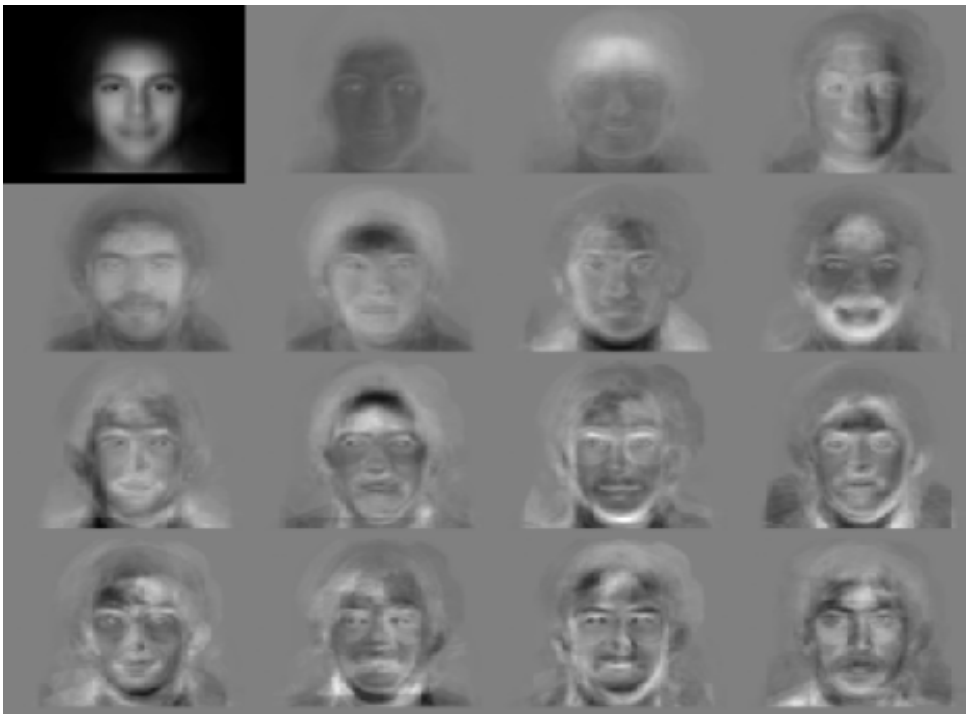

Figure 2 Feature Vectors Derived using 'eigenfaces'

1.3.2 LDA or Linear Discriminant Analysis

LDA is a statistical approach for classifying unknown classes using the training samples from known classes. This method maximizes the inter-class variance while minimizing the intra-class variance. The whole purpose of LDA is to classify 'objects' into 'groups' based on some distinguishing common 'features' which describe that 'object'. In general, an object is assigned to one of a number of predetermined groups based on observations made on the object.

When dealing with face data of high dimensionality, this technique faces the small sample problem. This problem arises because there are a small number of available training samples as compared to the dimensionality of sample space.

1.3.3 SVM or Support Vector Machine

A support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks (Support Vector Machine). Thus, when we have a set of points belonging to two classes, a SVM finds that hyperplane which separates the largest possible fraction of points belonging to a particular class on one side, while maximizing the distance of the other class from the hyperplane.

PCA is first used to extract the face features from the images and then the differentiating functions between each pair of images are learned by SVMs.

1.3.4 EBGM or Elastic Bunch Graph Matching

All human faces basically share the same topological structure. In this method, the faces are represented as connected 'graphs' with the nodes being reference points such as the eyes, the nose, the ears, etc. Each such 'node' consists of a number of complex Gabor wavelet coefficients called 'jets'. A 'labeled graph' is a set of nodes connected by edges. The nodes are labeled with jets and the edges with distances. Recognition is based on these labeled graphs.

EBGM relies on the concept that faces have non-linear characteristics which are not addressed by linear methods of analysis. A Gabor wavelet transform creates the architecture which projects the face onto an elastic grid. The Gabor 'jet' is a 'node' which describes the behaviour around a given pixel. Recognition is based on the similarity of the Gabor filter's response at each Gabor node.

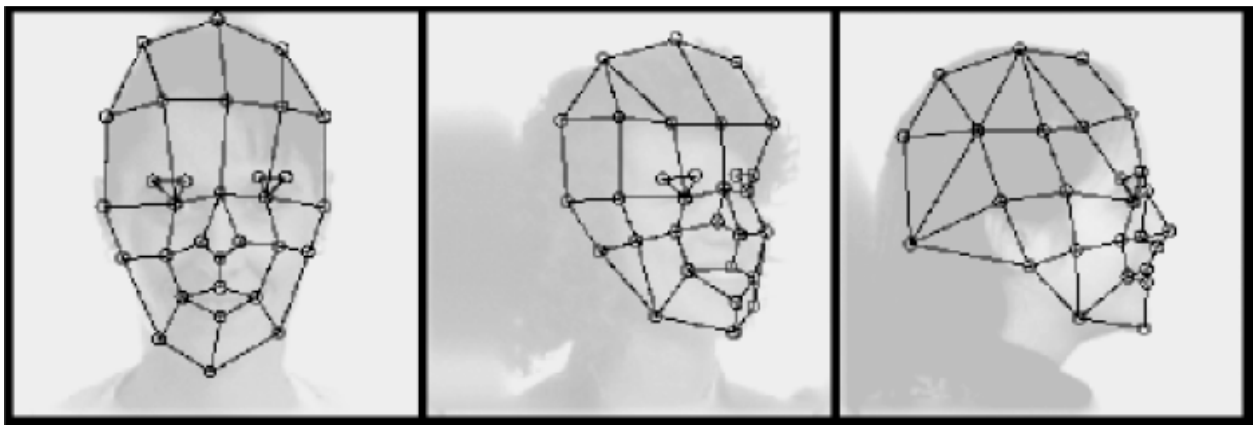

Figure 3 Transforming into a graph in EBGM

1.3.5 Trace Transform (Petrou & Kadyrov)

The 'Radon transform' computes projections of an image matrix along specified directions. The Trace transform is a generalization of the Radon transform. It consists of tracing an image using straight lines along which certain values of the image function are calculated.

A Trace transform is thus a function of the parameters of this 'tracing line'. In the plane, this line is characterized by two parameters.

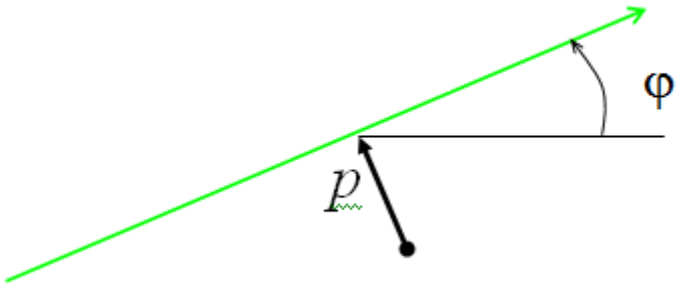

Figure 4 Characteristics of line, p, θ

The Trace transform can be used for face recognition efficiently as it is robust to rotation, translation and scaling. It can be used to calculate features from a facial image according to the tracing lines and then be matched with the standard for recognition.

Chapter – II

Fundamentals

In order to understand all the operations performed on the images, sufficient mathematical background regarding image enhancement, mathematical morphology and image transforms is required. This chapter is aimed at providing the required introduction to these useful tools so as to facilitate smooth understanding of the subsequent work.

2.1 The digital image and its neighbourhood operations

A digital image, ***f(m,n)*** may be described as a two-dimensional discrete space, where ***m,n*** denote the spatial coordinates and ***f*** denotes the amplitude at that particular coordinate, known as 'gray level' or 'intensity'. This discrete space is usually obtained from an analog image, which is a two-dimensional continuous space, through the processes of 'sampling' and 'quantization'.

The digital image is made up of a finite number of elements, known as 'pixels' or picture elements, which have both a spatial location and an amplitude value.

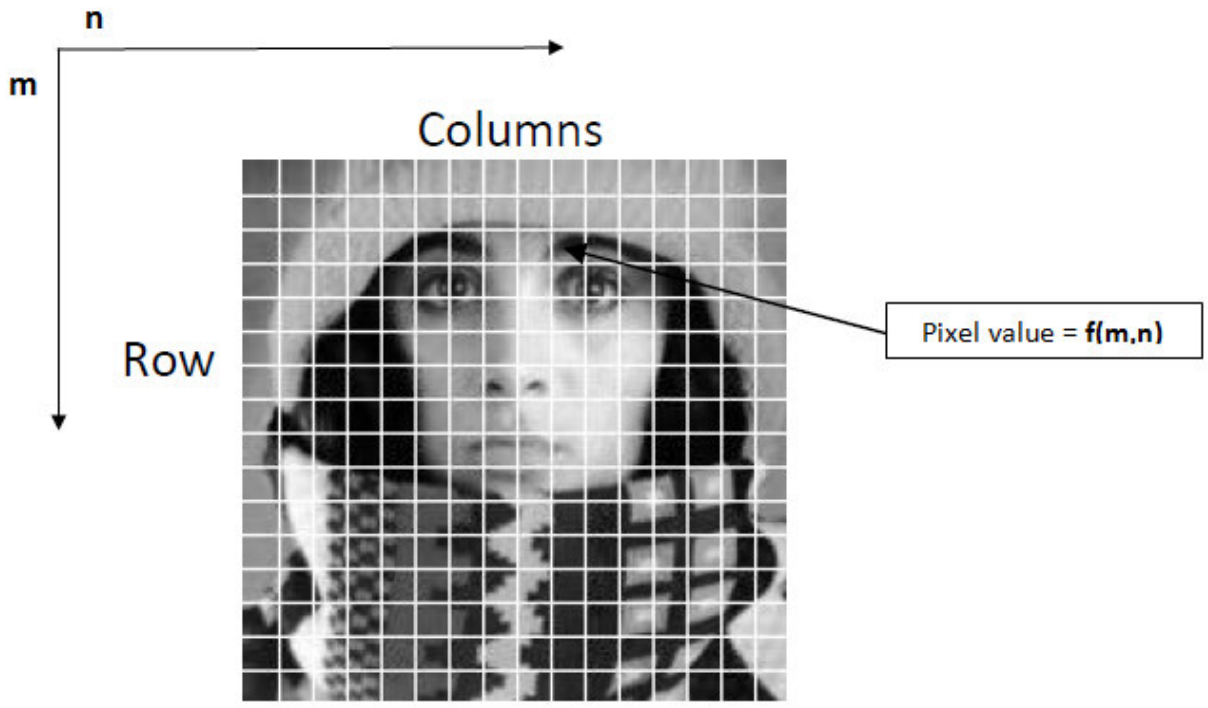

Figure 5 A digital image and its parameters

The neighbourhood of a point is defined as a set of pixels around the specified point subject to some rules. A pixel ***p(x,y)*** has four vertical and horizontal neighbours whose coordinates are given by –

$$(x+1,y),\ (x-1,y),\ (x,y+1),\ (x,y-1)$$

This set of pixels is called '4-neighbours of ***p***' and is written as $N_4(p)$.

The four diagonal neighbours of ***p(x,y)*** have coordinates –

$$(x+1,y+1),\ (x+1,y-1),\ (x-1,y+1),\ (x-1,y-1)$$

These pixels form a set which is denoted by $N_D(p)$. These points, together with $N_4(p)$ form the '8-neighbourhood of ***p***' and is written as $N_8(p)$.

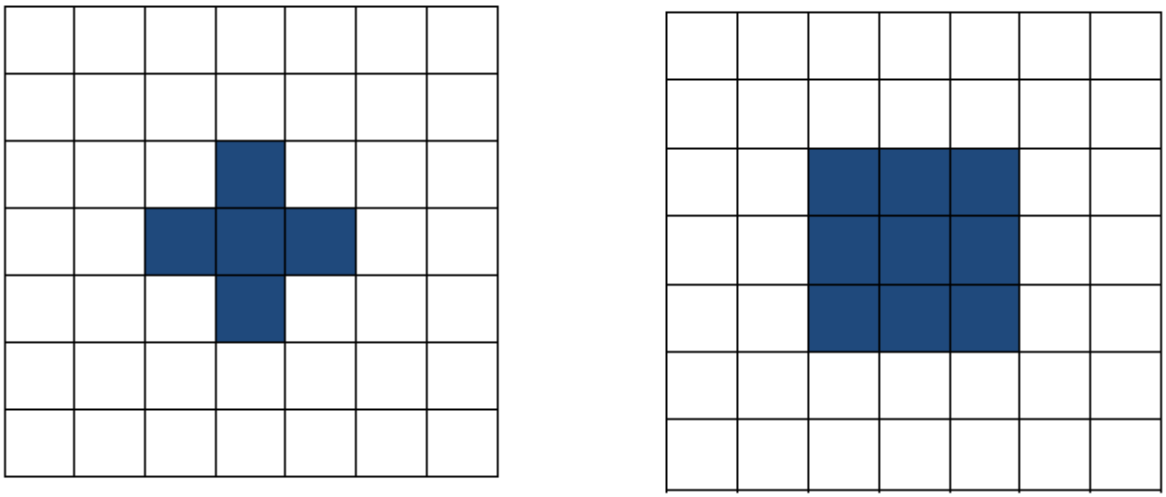

Figure 6 4-neighbourhood and 8-neighbourhood

2.1.1 Operations

The operations done on a digital image basically convert an input image to an output image with enhanced features which are to be extracted for some purpose. The types of operations can be majorly classified into –

1. Point operations – in which the output value of a pixel at a specified point is only subject to its input value.
2. Local operations – in which the output value of a pixel is dependent on the neighbourhood, defined around that pixel coordinates.
3. Global operations – in which the output value of a specific point is dependent upon all the values in the input image.

Mainly, neighbourhood based local operations are used.

2.1.2 Connectivity and CCA

After a suitable introduction to what the 'neighbourhood' of a pixel means, it is essential to know another important term, 'connectivity'. Two pixels are said to be '4-connected' or '8-connected' if there exists a path from one pixel to another comprising entirely of pixels which are in the 4-neighbourhood or 8-neighbourhood of each other.

0	0	255	0	0
0	0	255	255	0
0	255	255	255	0
0	0	0	255	0

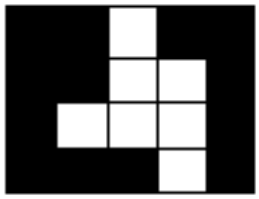

Figure 7 Example of a 4-connected region

In gray-scale images, the connected areas are found out by linking pixels which have values within a given set.

'Connected components analysis' or 'CCA' is a method of detecting connected regions in binary digital images. In raster scan mode, the binary image is searched, and a unique ID is labeled to the group of the pixels that neighbor each other.

Each connected area is denoted by a 'bounding box' whose vertex coordinates are stored in an array.

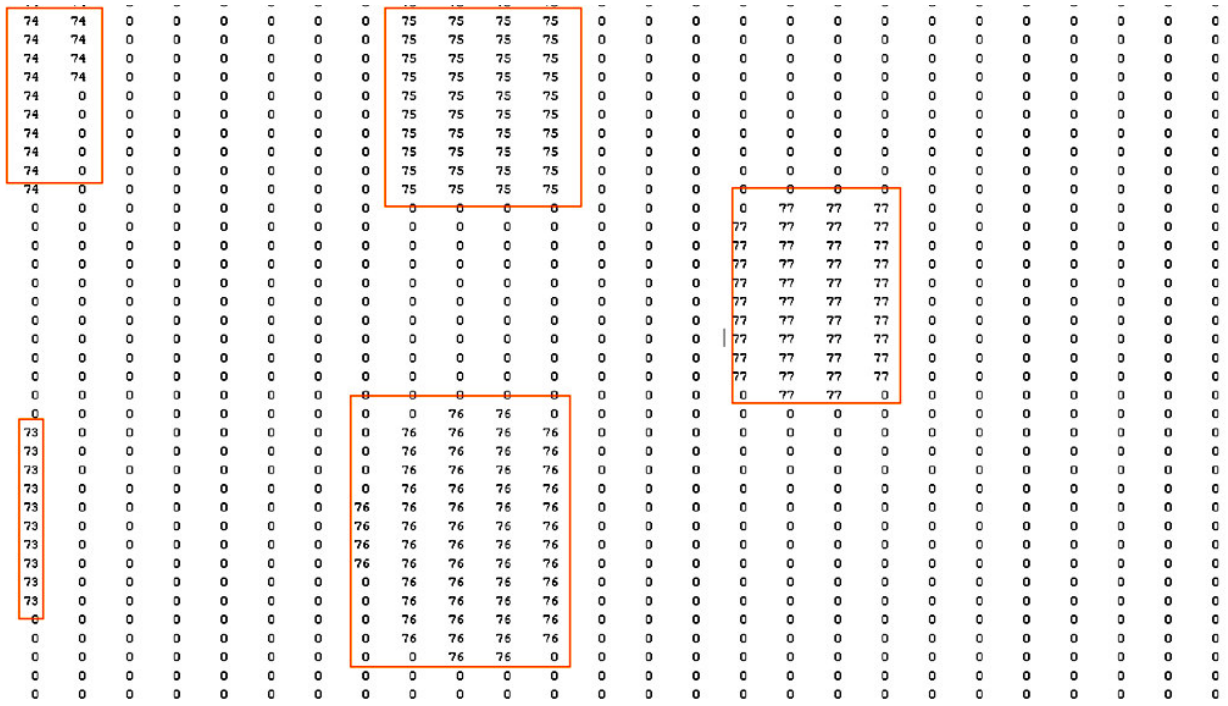

Figure 8 Portion of an image after CCA. The boxes show all the connected areas which are represented by the pixel ID's as shown

These bounding boxes may then be analyzed and other operations performed on them.

2.2 Image Enhancement Tool: Contrast Stretching

To incorporate images poor lighting conditions, the preprocessing consists of a variance based contrast enhancement operation. The variance of the image is calculated and checked against a cut-off value. If the variance is less than the threshold, then the image is taken under poor lighting conditions and must be corrected.

The contrast enhancement of the image may be done globally or adaptively. In adaptively enhancing an image, a window is convolved with the image and the local variance of the window is checked against the global variance (or a fraction of it) and accordingly the values adjusted. The contrast stretching function used is a simple piecewise linear function, although sigmoid functions may also be used.

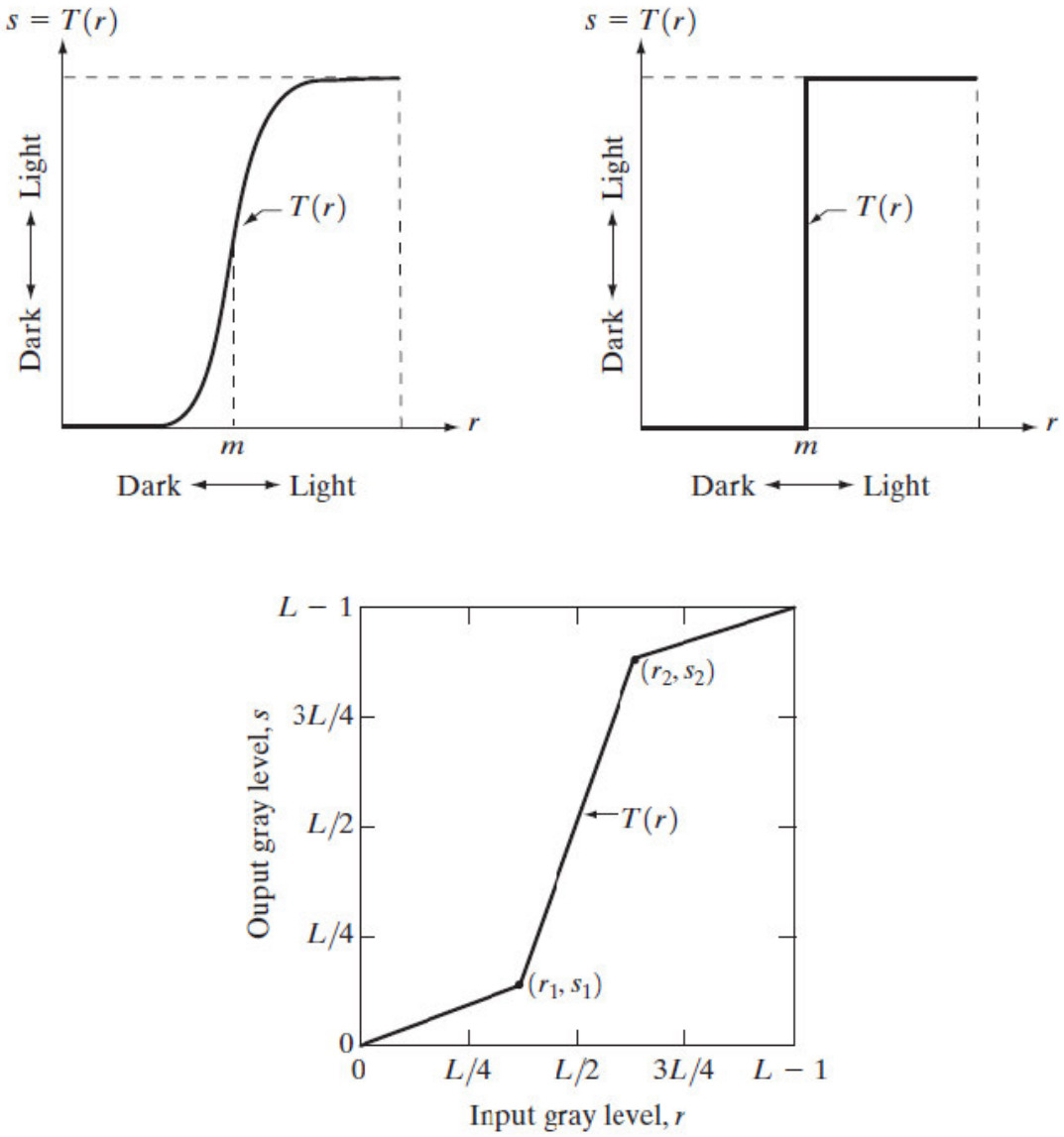

Figure 9 Functions used in contrast stretching, clockwise from top: Sigmoid function, Hard limiter, Piecewise linear

Figure 10 Result of adaptive contrast stretching: original (left) and enhanced (right)

2.3 Mathematical Tools

Certain tools and operators are central to the processing of digital images. These tools include 'convolution', 'morphology' and various 'transforms' (with our interest being in particular in the Gabor wavelet transform).

2.3.1 Convolution

The convolution of two signals to produce a third output signal is given symbolically by,

$$c = a \otimes b = a * b$$

Equation 1 Convolution of two functions *a* and *b*

In two dimensional continuous space, it is represented as,

$$c(x, y) = a(x, y) * b(x, y) = \iint a(\chi, \xi) b(x - \chi, y - \xi) d\chi\, d\xi$$

Equation 2 Convolution in continuous 2D space

In discrete space,

$$c[m,n] = a[m,n] \otimes b[m, n] = \sum\sum a[j,k] b[m- j,n - k]$$

Equation 3 Convolution in discrete 2D space

2.3.2 Mathematical Morphology (Gonzalez & Woods)

'Mathematical morphology' is used as a tool for extracting image components which have meaningful information, such as object boundaries, skeletons, convex hull, etc, and are, thus, useful in representing and describing region shapes.

2.3.2.1 Dilation

A and ***B*** being sets in $\boldsymbol{Z^2}$, the 'dilation' of ***A*** by ***B*** is given as,

$$A \oplus B = \{z \in E | (B^s)_z \cap A \neq \varnothing\}$$

Equation 4 Dilation

B^S signifies a symmetrical dilating function. It is also commonly known as a 'structuring element. This structuring element, B, is first modified by obtaining its reflection about its origin and translating this reflection by z. The dilation of A by this modified element is the set of all displacements z such that the modified element and A overlap by at least one element.

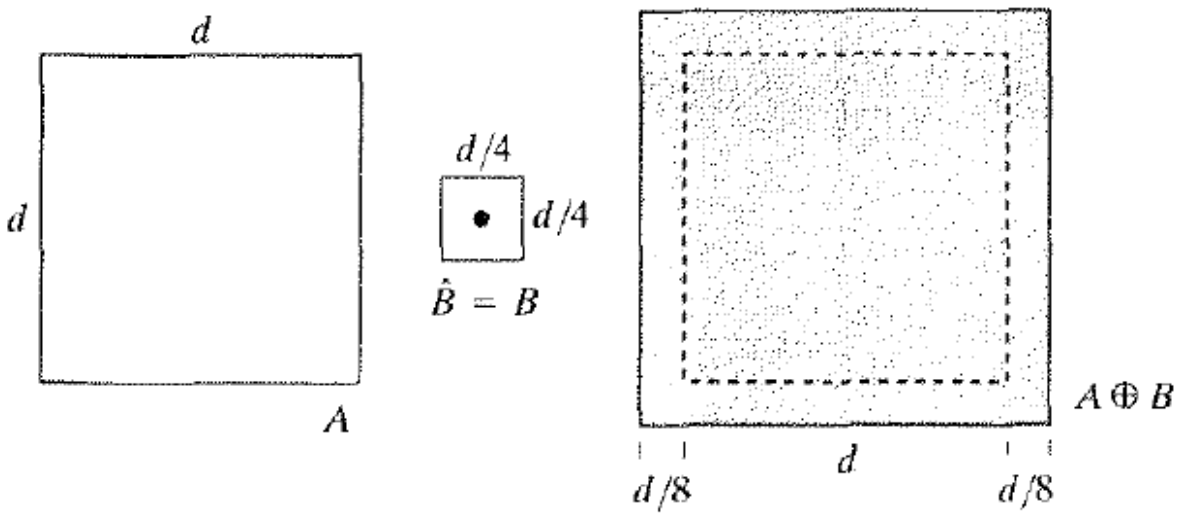

Figure 11 Shows the effect of 'dilation'

2.3.2.2 Erosion

For sets A and B, erosion is defined as,

$$A \ominus B = \{z \in E | B_z \subseteq A\}$$

Equation 5 Erosion

In words, this equation means that the erosion gives all translations z such that B translated by z is contained in A.

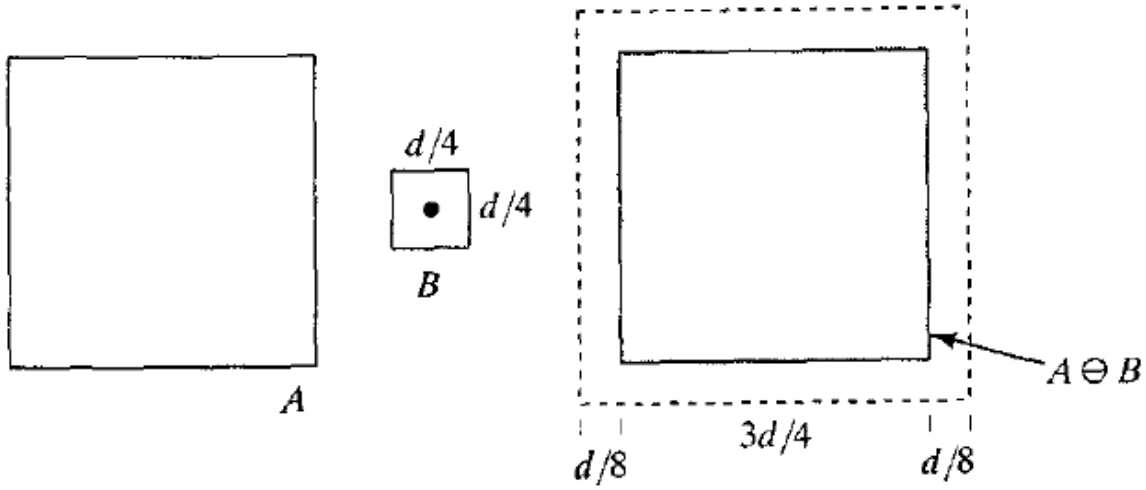

Figure 12 Shows the effect of 'erosion'

2.3.2.3 Opening

As observed, dilation expands an image and erosion is responsible for shrinking it. By combining these two operators a third operator, 'opening' is formed. Opening generally is used to break the narrow connections between two bounding boxes in order to get separate bounding boxes for better analysis. It is given by,

$$A \circ B = (A \ominus B) \oplus B$$

Equation 6 Opening

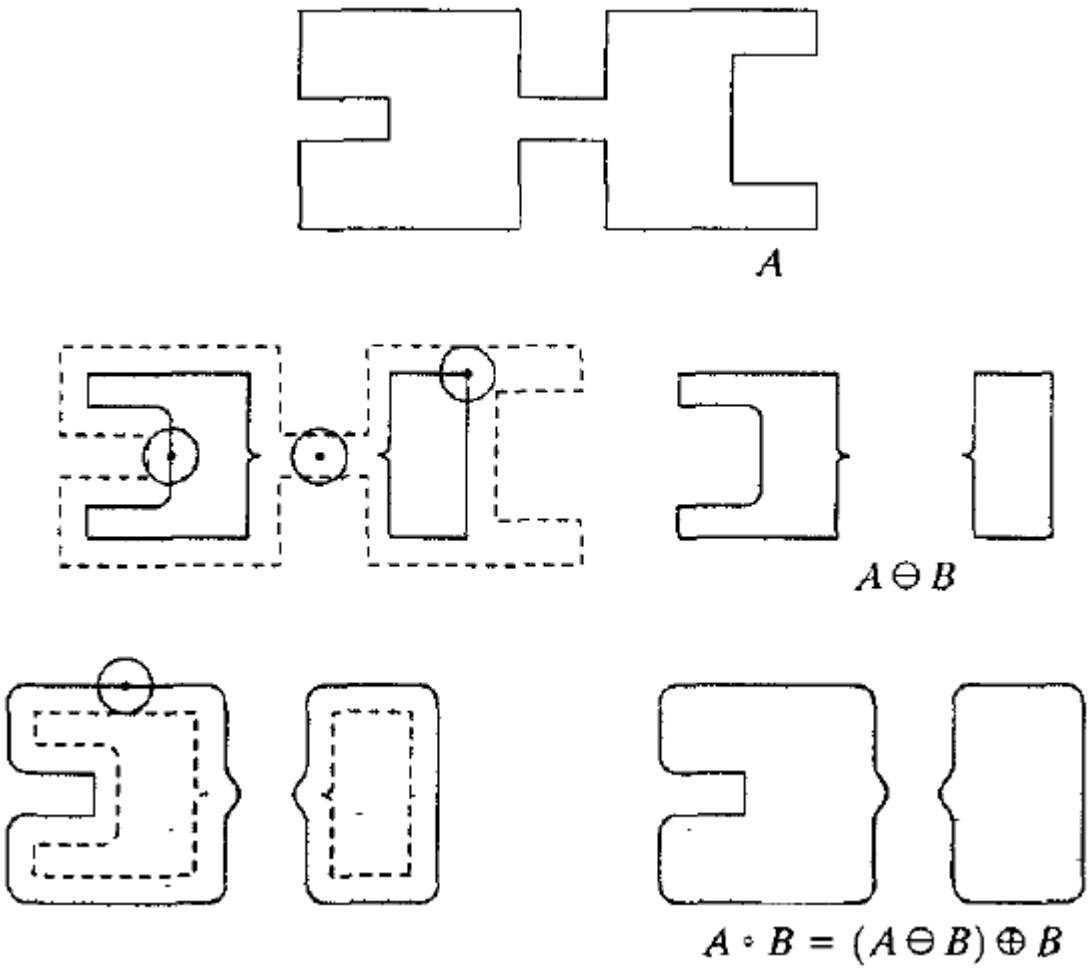

Figure 13 An example of 'opening'

As it can be observed, the 'opening' process has resulted in the separation of the object ***A*** into two components, which were earlier joined in the middle by a narrow isthmus.

2.3.2.4 Hole filling

A 'hole' may be defined as a background region which is encompassed by a connected border of foreground pixels.

Denoting by ***A***, a set whose elements are connected boundaries, each boundary enclosing a background region (or 'hole'), and taking sample points inside each hole, the objective is to fill all the holes.

The process is started by making an array of zeroes, X_0, which is of the same size as the array containing ***A***, except at the locations in X_0 corresponding to the sample point in each hole, which is set to the value with which the hole is to be filled (255 for binary image). The following is the iterative procedure to fill the holes,

$$X_k = \left(X_{k-1} \oplus B\right) \cap A^c$$

Equation 7 Hole filling

The termination condition is when $X_k = X_{k-1}$. The set X_k contains all the filled holes and the union of X_k and ***A*** contains all the holes and their boundaries.

The dilation would otherwise fill up the entire area but for its intersection at each step with A^c, the complement of ***A***. This is a typical example of how a morphological operation can be conditioned to meet a desired property and thus, it is called 'conditional dilation'

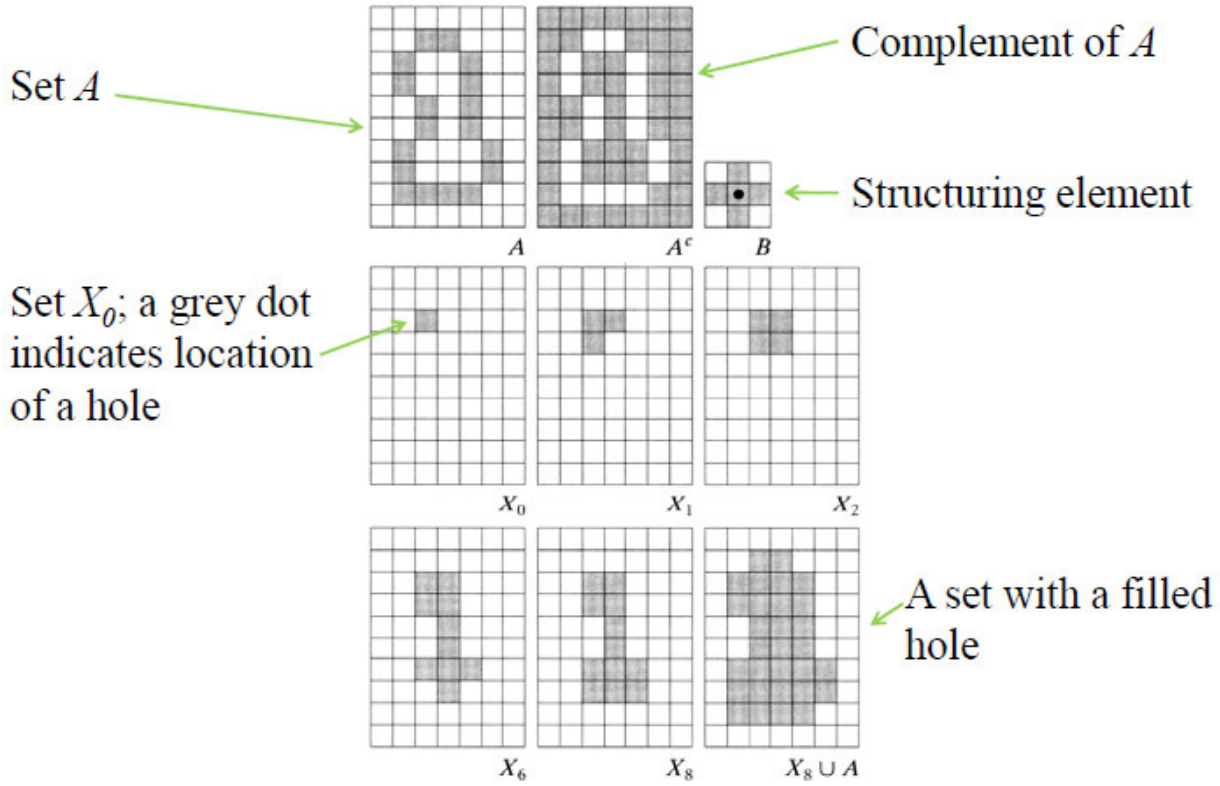

Figure 14 The process of 'hole-filling'

2.3.3 Image Transforms: Gabor Filter and Gabor Transform

The Gabor Filter, after Dennis Gabor, is a linear filter used mainly for edge detection. The frequency and orientation of these filters are closely similar to those of the human visual cortex and thus, they have their application in texture representation and differentiation.

A complex 2-D Gabor Filter is given by,

$$g(x,y) = \exp(-\pi(a(x-x_0)_r)^2 + (b(y-y_0)_r)^2) \,.\exp\,(j(2\pi(u_0x + v_0y) + P))$$

Equation 8 Gabor Filter in 2D

The subscript $_r$ stands for rotation operation,

$$(x - x_0)_r = (x - x_0)\cos\theta + (y - y_0)\sin\theta$$
$$(y - y_0)_r = -(x - x_0)\sin\theta + (y - y_0)\cos\theta$$

Equation 9 Rotated parameters in Gabor filter

Thus, the two-dimensional Gabor function is a convolution of a two-dimensional Gaussian function and two-dimensional sinusoid.

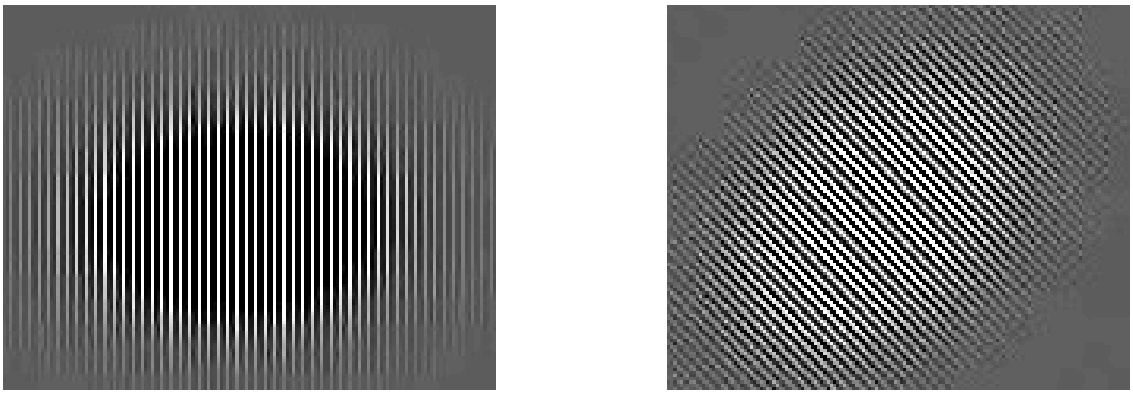

Figure 15 Gabor filters aligned at *0* and *45* degrees respectively, used for filtering oriented edges

The Gabor transform is a special case of the 'short-term Fourier transform'. It determines the 'frequency' and the 'phase' component of local sections of a signal as it changes over time. The Gabor transform of a signal ***x(t)*** is given by,

$$G_x(t, f) = \int_{-\infty}^{\infty} e^{-\pi(\tau - t)^2} e^{-j2\pi f\tau} x(\tau)\, d\tau$$

Equation 10 The Gabor transform

2.3.4 Euler number (Saveliev)

If a binary image of any dimension is represented by its 'cell decomposition'. Then the 'Euler number', or 'Euler Characteristic', of the image is stated as the 'alternating sum of the numbers of cells of each dimension'.

In 2D images, it is,

$$e = \textbf{Number of vertices} - \textbf{Number of edges} + \textbf{Number of pixels}$$

By 'Euler-Poincare' formula, it may be simplified to give,

$$e = \textbf{Number of objects} - \textbf{Number of holes}$$

Equation 11 Finding out the Euler number

2.4 The HSV Colour Space

HSV colour space is a representation of the RGB colour space in a cylindrical coordinate system. The RGB colour space is basically a Cartesian representation of the colour space where the chroma and the luma component of the colour are non distinguishable. But transform to the HSV cylindrical colour space separates the chroma and luma component so that the image under consideration is not dependent on the uniformity of the diffused light source.

HSV stands for Hue, Saturation and Value, where the radius of the colour point from the central axis in the cylindrical coordinate system represents the Saturation, the angle subtended around the central axis is the Hue and the height of the colour point from the base along the central axis is the Value of a colour point.

Hence the projection of the point on the base 2-D plane represents only the chroma component of the colour and the luma component is separated and ignored.

- Hue: It is the representation of the colour of a pixel in HSV space. It encodes the chroma of the point.
- Saturation: It represents the degree of colour intensity of the colour representated by the Hue value.
- Value: It represents the intensity/luma of the colour point and therefore is independent of the colour intensity of the point.

The HSV colour space magnitudes can be generated for a colour point if the RGB colour space magnitude is known, using the following transformations:

$$H = \arccos \frac{0.5*(2R-G-B)}{\sqrt{((R-G)^2-(R-B)(G-B))}}$$

$$S = \frac{max(R,G,B)-min\,(R,G,B)}{max\,(R,G,B)}$$

$$V = max(R,G,B)$$

Equation 12 Conversion to HSV from RGB

Chapter – III

Detection

The process of identification of faces begins with first detecting it from the image with the face against a simple background or a complex scene. The algorithm presented here is one which detects faces, and crops them to be fit for recognition, from images of complex scenes with multiple faces.

3.1 Skin Segmentation

After preprocessing of the image to enhance the contrast, it is subjected to a skin colour based segmentation process to separate the faces from the background regions.

Skin is one of the most distinguishing properties of the human face surface. Skin colour is quite a concentrated and stable region in images. Using this useful clustering data provided by the chrominance (colour) information of the skin, it can be distinguished from the background in order to locate the possible areas which might contain the face.

The HSV (Hue, Saturation and Value) colour space is chosen for segmentation since it decouples the chrominance information from the luminance information. Thus we can only focus on the hue and the saturation component (Mohsin, Ahmed, & Mar, 2003). The faces in each training image were extracted their histogram, plotted for their 'H' and 'S' colour component, were averaged to give two final averaged histograms as shown below. The histograms show that the 'H' and 'S' components for faces are properly clustered. This data was used to define suitable thresholds for 'H' and 'S' space that correspond to faces. These threshold values were fit into the segmentation routine.

Best fit Gaussian functions are then calculated for each histogram, and the mean (value corresponding to the highest point of the Gaussian) and variance (taken, in this case, as the period which corresponds to values above 50% of the highest point) are found out. Skin segmentation is then carried out by checking if the 'H' and 'S' values of an input image in HSV space falls within the period denoted by the variance found out from the corresponding histogram.

If the chrominance properties are found to be unsatisfactory, the pixel at that particular spatial position is changed to black, else kept same as before. Thus, the output image contains only the faces and the face-like regions. This image further tested and false regions rejected to finally give the face regions.

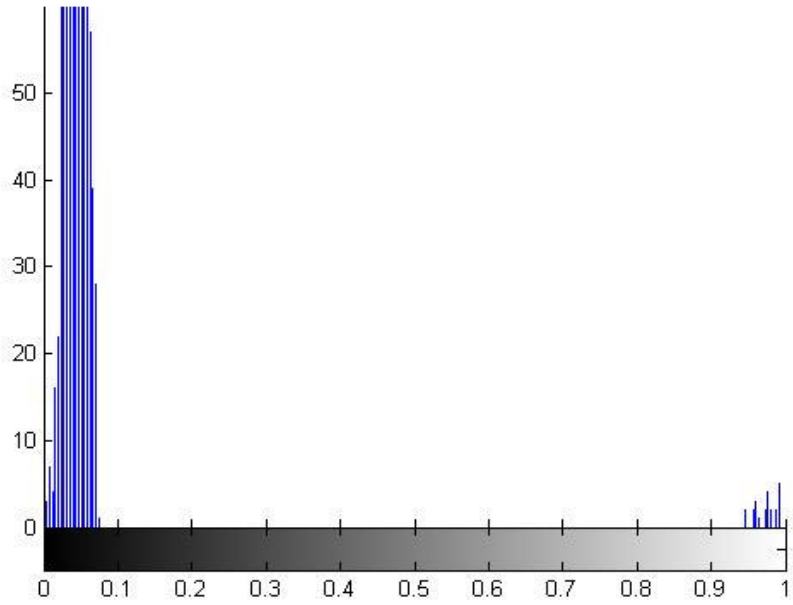

Figure 16 Average histogram for the 'hue' component

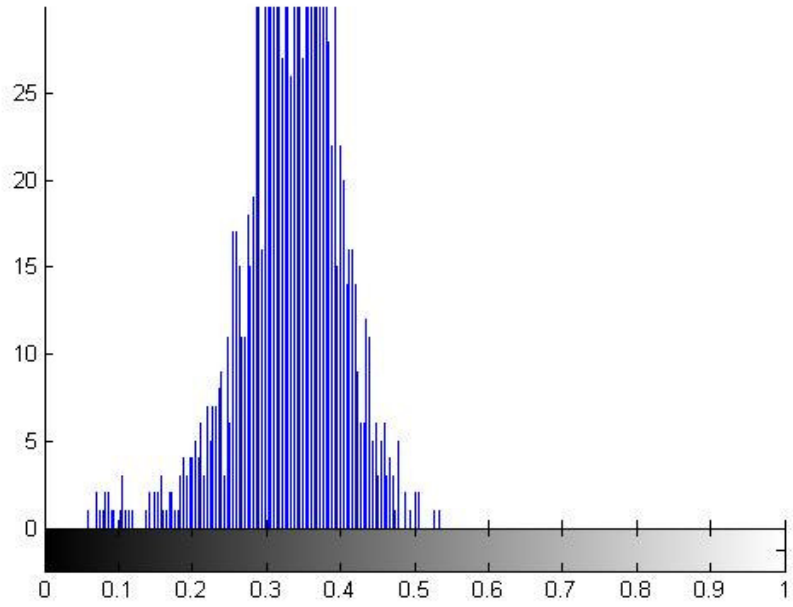

Figure 17 Histogram for the 'saturation' component as obtained from the training face images

Figure 18 Original image

Figure 19 Image in HSV space

Figure 20 After histogram based skin segmentation

3.2 Removal of unwanted regions and noise

3.2.1 Morphological Cleaning

After the histogram based segmentation, it is observed that all the non-skin regions have been rejected from the image. However, the image is still noisy and cluttered. The image is then subjected to a series of morphological operations, which are performed sequentially to 'clean up' the image. The end objective is to get a 'mask image' that can be put on the input image to yield face regions without any noise and clutter.

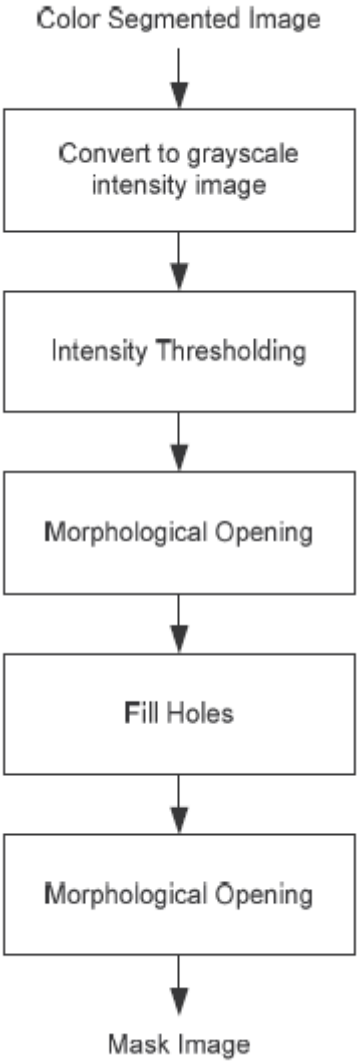

Figure 21 Sequence of steps to 'clean' the image

The sequence of steps is further described as follows:

1. The colour image is first converted into a gray scale image since morphological operations are known to easily work on intensity and binary images.

2. ‘Intensity thresholding’ is done to split up dark regions into smaller regions so that they can be effectively cleaned up by ‘morphological opening’. The threshold is set low enough so that it does not damage parts of a face but only create holes in it.

3. Morphological opening is then performed to eliminate very small objects from the image while keeping the shape and size of larger objects in the image unbroken. A disk shaped structuring element of radius 1 is used.

```
0   1   0
1   1   1
0   1   0
```

Figure 22 Disk shaped structuring element of unity radius

4. ‘Hole-filling’ is performed to maintain the face regions as single connected regions to account for a second morphological opening with a much larger structuring element. Or else, the mask image will contain many cavities and holes in the faces.

5. Morphological opening is again performed to remove small to medium sized objects which can be safely neglected as non-face regions. This time a bigger structuring element, disk shaped with radius 8, is used.

Figure 23 Image after morphological operations have been performed

Figure 24 Mask image applied to original

3.2.2 Connected Regions Analysis

3.2.2.1 Rejections based on Geometry

The faces are generally contained in rectangular moderate sized bounding boxes. Thus, four classes of regions are defined which have a high probability of being non-faces based on their bounding box:

- 'narrow' regions which have a small width
- 'short' regions which have a small height
- 'narrow and tall' regions which have a small width but large height
- 'wide and short' regions which have a large width but small height

'Wide and tall' regions are not counted. Classes such as that might contain multiple faces in them.

Based on the training set face images, we know the approximate size of the face bounding box. Also, the minimum size of the bounding box is determined by the capability of the image identifier to which the cropped face images will be finally sent. Thus, the thresholds were calculated for each class. The algorithm was then applied as follows:

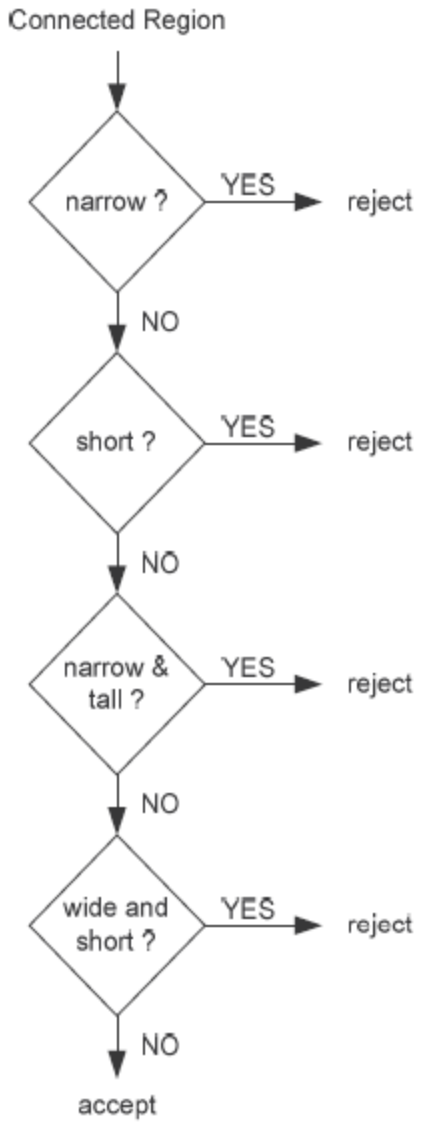

Figure 25 Sequence of steps for 'rejection based on geometry'

The resultant image contains regions which are all skin-coloured and of dimensions suitable for containing a face fit for the recognition process.

3.2.2.2 Euler Characteristic based Rejection

Euler number analysis is based on the information that the facial regions corresponding to the eyes, the nose and the lips are quite darker as compared to the other face regions and they show up as 'holes' after proper binarizing via thresholding. The Euler numbers for these binarized regions are then calculated.

An adaptive system is used to produce the threshold for each connected area. First, the mean and the standard deviation of the intensity level of the region are calculated. If there is a large spread and the ratio of mean to standard deviation is high, the threshold is set to a fraction of the mean. This stops darker faces from splitting up into many connected regions after thresholding. Otherwise, the threshold is set to some multiple of the standard deviation to make sure bright faces are accounted for.

After computation of the Euler number $\boldsymbol{e}$. If $\boldsymbol{e} \leq 0$ (i.e. less than two holes) the region is rejected. This is based on the fact that the face has at least two holes due to the presence of the eyes and hence the Euler number for a face region must be less than zero.

The result of this filtering process leaves us with the following image:

Figure 26 Image contains less noise after Euler number based filtering

3.3 Template Matching

It may so happen that one particular region contains more than one face, or face with portion of the background with a skin-like tone. In these situations, it is difficult to extract the images directly. This problem has motivated the process of 'template matching' where an average image is convolved with the detected areas to give all the faces which are present.

The average template was constructed by averaging all the training set images,

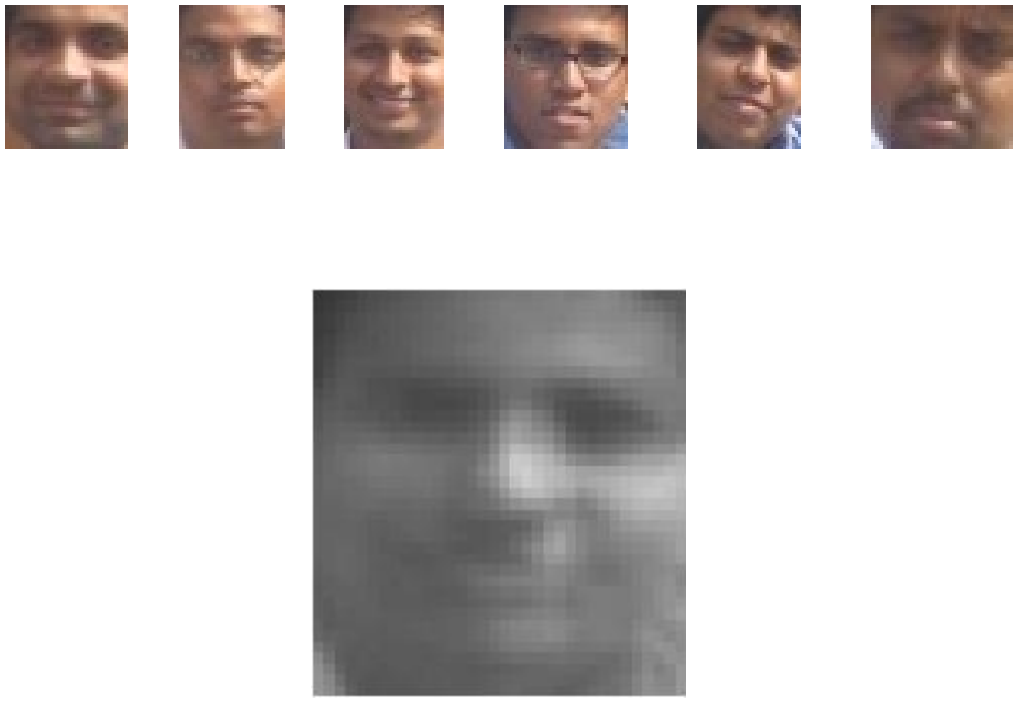

Figure 27 (top) Training images and (bottom) Average image template

This template was convolved repeatedly with the image and the maximum peaks were extracted until the peak failed to exceed a calculated threshold value. The coordinates of the centre of the bounding boxes which matched the most with this average template were stored in an array.

In every run, the region with the highest match was blacked out to facilitate the finding out of the region with the next highest match and so on.

The results of the template matching algorithm are the coordinates of the faces with the size of the region being same as that of the average image.

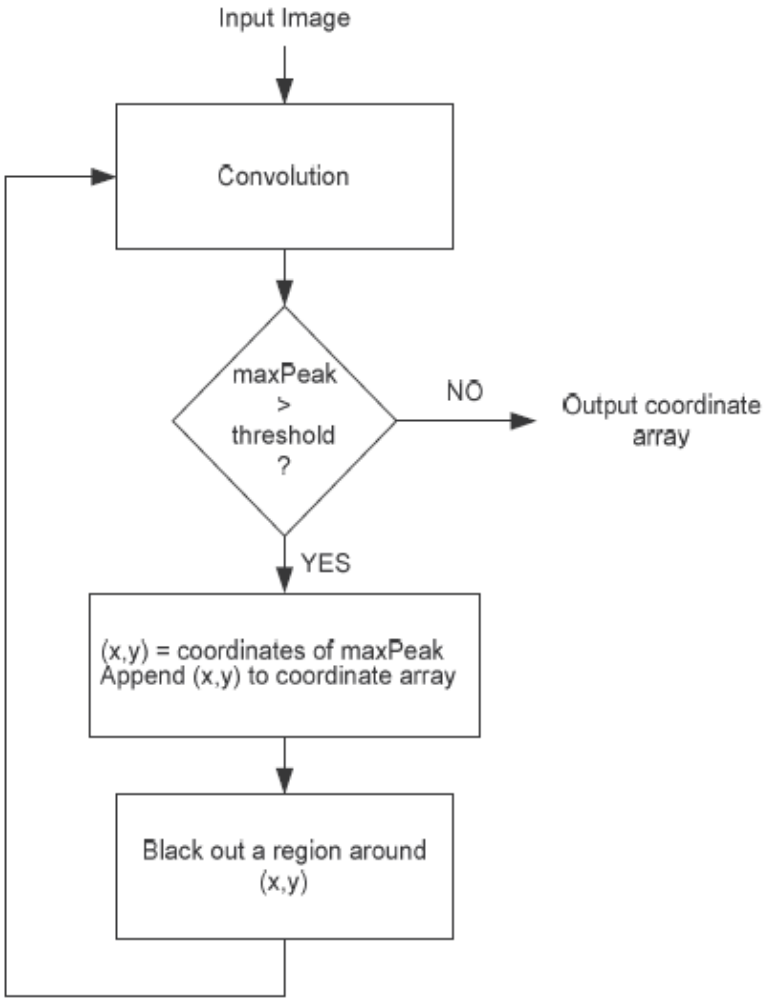

Figure 28 Algorithm for template matching

3.4 Results of the Detection Process

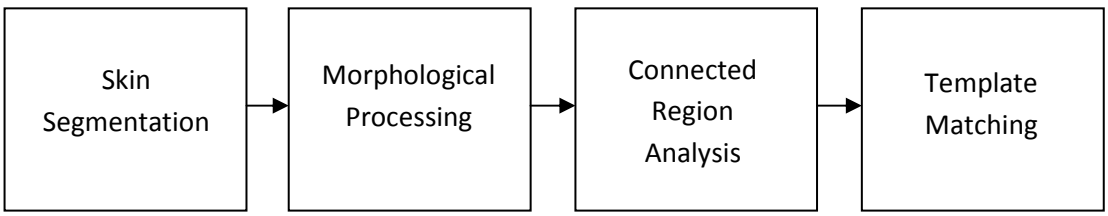

Figure 29 Detection process block diagram

Above is the block diagram of the complete detection process. The results are shown as below,

Figure 30 Original

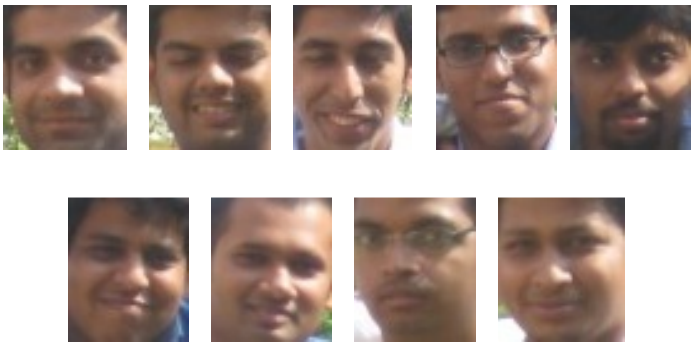

Figure 31 Detected faces

Chapter – IV

Recognition

Once the detection phase is complete, the recognition phase begins. Ideally, the cropped images of the extracted faces from the face detector are to be fed into the identifier. However, since a proper database of faces under controlled conditions could not be constructed, the algorithm developed was tested on existing databases acquired from the internet.

4.1 Elastic Bunch Graph Matching or EBGM

The basic idea of EBGM is only implemented after the facial extraction segmentation is complete. Hence, we begin our analysis considering that the input database is now no more a set of images with facial regions but a set of segmented bounding box each signifying a facial image.

The facial image in each bounding box is assumed to be thresholded by skin segmentation in the extraction process such that only the skin region of the facial region is identified and rest all regions which include background, accessories and hair connected in the neighborhood of the image being thresholded to zero.

The main idea of the EBGM is to design an algorithm that can comparatively analyze two images and find a quantitative value for the similarity between them. (Jean-Marc Fellous, 1999)

The level of difficulty of analyzing the similarity of two facial segments of same person increases considering the distortions that may range as:

1. Position
2. Size
3. Expression
4. Pose

In our current analysis of the algorithm, we are assuming that the matching system is to be used as an integration of user identification systems, where there is a set of database images that are registered as user with a fixed background and position consideration. Hence, during analysis we are making a basic assumption that the image is not distorted in sense of position and size, as the capturing source (camera and processing equipment) is constant in all cases.

So the only variation/ distortion that have to be accommodated in the algorithm are for expression distortions and pose distortions:

- **Expression Distortions**: The facial expression of the person in the facial image is prone to change in every image hence cannot match the base database image.
- **Pose Distortions**: The position of the person with respect to the camera can vary a little in the 2-D plane of the image but not over a large range.

Also the color of hair, clothes and accessory can change frequently and will never remain constant to the base image. But it is of little consequence as the facial extraction is supplemented with skin segmentation, which remains constant irrespective of other changes.

4.1.1 Correlation between image and graph

The basic object representation used in case of EBGM is a graph. This graph is used to represent a particular face segment, is therefore a 'facial graph'.

The facial graph is labeled with nodes and edges. In turn the nodes are identified with wavelet responses of local jets and the edges are identified by the length of the edge.

- **Node:** It is a point in the 2-D facial graph of the image that signifies the phase and the magnitude of the wavelet response of the facial image in the locality of the node point.
- **Edge**: It is a line that connects the nodes. Every two node in the graph is interconnected with an edge, which is represented with the magnitude of the length of the edge.

Thus a facial graph is a set of nodes that are related to certain facial regions as decided by the user and a set of corresponding edges. These constitute a basic structure of the facial graph, which is sufficient to discriminate among other non-facial graph that will have a different structure. (DeValois, 1988)

But in case of classification among different image graph, the basic structure is not sufficient and further information is calculated that is distinguishable amongst facial graphs.

4.1.2 Preprocessing (Gabor Wavelets)

The image after segmentation is in RGB space, where the facial region is expressed by their RGB values. But the RGB space is extremely prone to changes in illumination and slight movement or distortions. Hence, the facial segment is converted to wavelets using wavelet transform.

Gabor wavelets generated using Gabor Wavelet Transform is Gabor filters applied to wavelet space. They are predominately edge detection filter that assigns magnitude and phase depending on edge directions and intensity in varying dilation and rotational values.

They are mainly implemented by convolving a function with the image to generate Gabor Space. (Daugman, 1988)

Thus for a set of degree of rotations and dilations a set of Gabor kernels are generated. These kernels will extract the 'jets' from the image.

These 'jets' are defined as a small patch of grey values in the wavelet transformed image around a given pixel $\boldsymbol{x = (x,y)}$.

The small patch of grey values is the result of wavelet transform convolution:

$$\mathcal{J}_j(\vec{x}) = \int \mathcal{I}(\vec{x}')\psi_j(\vec{x} - \vec{x}')d^2\vec{x}'$$

Equation 13 Convolution with Gabor Kernels to generate wavelet transformed image

The convolution function is in turn a Gabor Kernel convoluted over the 2D image. This 'Gabor kernel' is represented as,

$$\psi_j(\vec{x}) = \frac{k_j^2}{\sigma^2} \exp\left(-\frac{k_j^2 x^2}{2\sigma^2}\right) \left[\exp(i\vec{k}_j\vec{x}) - \exp\left(-\frac{\sigma^2}{2}\right)\right]$$

Equation 14 Family of Gabor Kernels for *j* varying from 0 to 39

Here $\boldsymbol{k_j}$ is the wave vector that is restricted by the Gaussian envelope function. For our calculations, 5 different set of frequencies for index $\boldsymbol{\gamma}$ = 0, 1…4 and 8 sets of orientation directions $\boldsymbol{\mu}$ = 0, 1, 2….7 are taken.

$$\vec{k}_j = \begin{pmatrix} k_{jx} \\ k_{jy} \end{pmatrix} = \begin{pmatrix} k_\nu \cos\varphi_\mu \\ k_\nu \sin\varphi_\mu \end{pmatrix}, \; k_\nu = 2^{-\frac{\nu+2}{2}}\pi, \; \varphi_\mu = \mu\frac{\pi}{8},$$

Equation 15 Wave vector *kj* for *j* varying from 1 to 39

The index number $\boldsymbol{j = \mu + 8\gamma}$ is the sampling that covers the whole Gaussian space for 40 values of $\boldsymbol{j}$. The width $\boldsymbol{\sigma/k}$ is Gaussian controlled with $\boldsymbol{\sigma = 2\pi}$. The

second term in the bracket of the equation of the Gabor Kernel makes the equation DC-free and the integral $\int \psi_j(\vec{x}) d^2\vec{x}$ vanishes.

The preference of Gabor wavelet transform over normal edge detection and analysis is evident in this case as Gabor filters are much more robust to the data format of biological relevance which in this case is facial segments. Also the robustness is defined as the result of the transform is not susceptible to variation brightness when the Gabor wavelets are considered DC-free.

Figure 32 Gabor kernels for orientations μ = 0 and 1 at frequency ν = 0

Figure 33 Gabor kernels for orientations μ = 2 and 3 at frequency ν = 0

Figure 34 Gabor kernels for orientations μ = 4 and 5 at frequency v = 0

Figure 35 Gabor kernels for orientations μ = 6 and 7 at frequency v = 0

Figure 36 Gabor kernels for orientations μ = 0 and at frequency v = 1 and 2

Figure 37 Gabor kernels for orientations μ = 0 and at frequency ν = 3 and 4

If the jets are normalized then the Gabor wavelet transform is set immune to contrast variations. But the jets have limited control over the variation in form of rotation, translation and scaling. Though rotation and translation to a small degree does not affect the magnitude of jets but result in high phase variations.

Hence this phase variation can be ignored when analyzing images with certain degree of translational or rotational distortions or the phase can be used to calculate the degree of displacement between two images and compensate this distortion.

Though the use of large kernels is a disadvantage as it increases significantly the sensitivity of background variations, it is of little consequence in our case where the background (region except the facial skin region) is already thresholded to a minimum constant. (Potzsch, 1997) (Pollen, 1981)

4.1.2.1 Comparison between Jets

The basic structure of image graphs reduces the comparison between image pixels to comparison between the jets representing that particular pixel. Hence, the comparison between images is to assess the similarity in the structure of the face segments.

In an image wavelet two jets that represent two nearby pixels in its corresponding image, have different values of jets though they may represent the same local feature mainly due to phase rotation. Thus, if we ignore the phase of the jets, the magnitude of the jet remains similar for the nearby pixels representing same local feature.

Figure 38 Original image (right) and Gabor wavelet transformed image (left) for person 1 in database

Figure 39 Original image (right) and Gabor wavelet transformed image (left) for person 2 in database

Figure 40 Original image (right) and Gabor wavelet transformed image (left) for person 3 in database

Figure 41 Original image (right) and Gabor wavelet transformed image (left) for person 4 in database

Figure 42 Original image (right) and Gabor wavelet transformed image (left) for person 5 in database

It can be hereby inferred that a facial feature causes low variation in the magnitude of the jets irrespective of the degree of brightness and contrast variation.

The similarity function between two jets only on the basis of their magnitudes is,

$$S_a(\mathcal{J},\mathcal{J}') = \frac{\sum_j a_j a'_j}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}},$$

Equation 16 Similarity measure

When, $\boldsymbol{J_j = a_j\, exp(i\varphi_j)}$.

Here, $\boldsymbol{J_j}$ are the jets for the Gabor Wavelet transform at certain frequency and magnitude. (Lades, 1993)

If the jet ***J*** is taken as a fixed pixel position and the jet ***J'*** is varied by varying position pixel ***x***, then the similarity function is a smooth curve with low degree of variation in the neighborhood of ***J***. This shows the equivalence of the magnitude of the jets for similar local feature. The smooth curve forms shallow attractor basin with optima to indicate local similarity peaks.

But in the above comparison of jets, the phase data is ignored. But the phase values have its set of importance in feature matching as:

1. Similar magnitude patterns can be discriminated
2. The phase variation is a measure for accurate jet localization of a feature point.

So the inclusion of phase dependence is important to make the similarity function more robust to pattern variations in a more accurate manner. But in order to stabilize the phase sensitive similarity function, the compensation factor needs to be subtracted that nullifies the phase variation in nearby pixel points.

For the compensation factor it is assumed that the jets compared in the similarity function belongs to nearby point hence have a small displacement between them. Thus a small relative displacement ***d*** is implemented to generate following phase sensitive similarity function,

$$S_\phi(\mathcal{J}, \mathcal{J}') = \frac{\sum_j a_j a'_j \cos(\phi_j - \phi'_j - \vec{d}\vec{k}_j)}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}}.$$

Equation 17 Similarity Function *S* for jets including phase

4.1.2.2 Calculation of Displacement, *d*

In order to calculate the displacement factor, the phase sensitive similarity function is expanded in its 'Taylor series' form. Now as the displacement factor is valid for small displacements only, its value will be low and hence can be ignored for higher powers of Taylor series than 1.

The basic idea is to maximize the phase dependent similarity function and calculate the compensation required by the displacement factor ***d*** to nullify the difference. (Fleet, 1990) (Theimer, 1994)

$$\mathcal{S}_\phi(\mathcal{J},\mathcal{J}') \approx \frac{\sum_j a_j a'_j[1-0.5(\phi_j-\phi'_j-\vec{d}\vec{k}_j)^2]}{\sqrt{\sum_j a_j^2 \sum_j a_j'^2}}\ .$$

Equation 18 Taylor Series expansion of phase included similarity function

Now, after the expansion, solving for ***d***, the conditions are set as,

$$\frac{\partial}{\partial d_x}\mathcal{S}_\phi = \frac{\partial}{\partial d_y}\mathcal{S}_\phi = 0$$

Such that the equation gets reduced to,

$$\vec{d}(\mathcal{J},\mathcal{J}') = \begin{pmatrix} d_x \\ d_y \end{pmatrix} = \frac{1}{\Gamma_{xx}\Gamma_{yy}-\Gamma_{xy}\Gamma_{yx}} \times \begin{pmatrix} \Gamma_{yy} & -\Gamma_{yx} \\ -\Gamma_{xy} & \Gamma_{xx} \end{pmatrix} \begin{pmatrix} \Phi_x \\ \Phi_y \end{pmatrix}$$

$$\begin{aligned} \Phi_x &= \sum\nolimits_j a_j a'_j k_{jx}(\phi_j-\phi'_j), \\ \Gamma_{xy} &= \sum\nolimits_j a_j a'_j k_{jx}k_{jy}, \end{aligned}$$

Equation 19 Displacement Vector

$\boldsymbol{\Phi_y}$ $\boldsymbol{\Gamma_{xx}}$ $\boldsymbol{\Gamma_{yx}}$ $\boldsymbol{\Gamma_{yy}}$ are therefore defined accordingly.

This equation gives the displacement factor between the jets considering they belong to two neighboring pixel points in the locality. Thus the range of the displacement that can be calculated with it extends to half the wavelength of highest frequency kernel that can be up to 8 pixels for high frequencies.

The inclusion of the displacement factor can also iteratively point towards the jet that will maximize the similarity in the neighborhood. Once, the displacement factor start reducing in certain direction of position variation, further variations are concentrated on that direction until minima of the displacement factor is achieved, which is the point for high phase sensitive similarity.

4.1.2.3 Face graph Representation

A general image graph is made up of nodes and edges. Here, the node points are the bunch of jets quantifying the local feature using Gabor kernels and the edge is the distance between any two nodes.

The implementation of image graphs in facial image segments generates face graph representation or simply face-graphs.

For face graph generation from facial images, a set of 'fiducial points' are decided upon where each fiducial point represents a unique facial feature that will help in generating a representative face graph for that person.

For our analysis the fiducial point chosen were:

1. Iris of left eye
2. Iris of right eye
3. Nose tip
4. Upper Lip tip
5. Chin Tip

Hence a labeled graph will have $N = 5$ nodes at the above mentioned 5 fiducial points and $E = 10$ edges connecting between those points. Here an edge $e<E$ connects two nodes n and n'.

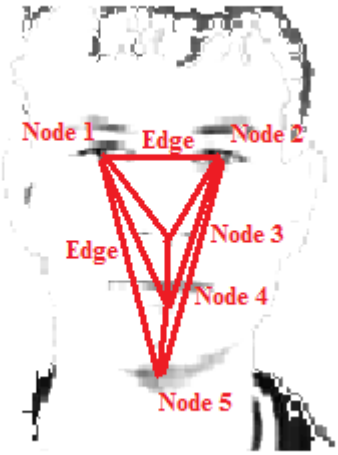

Figure 43 The face graph that can be generated with the considered set of fiducial points

Now the face graph generated from a facial image segment, manually or automatically is object-adapted as they represent features of the object (in this case person) in the facial image.

4.1.3 Face Bunch Graphing

In case of generation of algorithm for face identification, the database of the facial image segment will be huge. Any new input image will generate a set of facial image segments that needs to be compared with a fixed database of face graphs to find a face match.

Thus, the fiducial point identification cannot be done for all the facial segments manually. Hence a general representation is to be generated for wide range of variations of facial characteristics like different shapes of eyes, mouth and noses as they form the basic set of fiducial points in our analysis.

But generating a separate face graph for each feature will create an excess database, which can be reduces by bunching data into a facial graph. Hence, a 'face bunch graph' is generated from the individual set that will have a stack of jets at each node that represents a fiducial point. (Beymer, 1994)

In our algorithm, that essentially matches every new facial input to a set database of faces to find a facial match to grant access, we have low degree of translation and rotational error.

Hence the face bunch graph is generated for each individual person. That is, for each person a set of facial segment is used as model image set and a face bunch graph is generated that represents uniquely the facial characteristics of that person.

This face bunch graph has a set of jets from extracted from each model image representative of a particular person. These jets are stacked at each fiducial node and the edges of each model face graph are averaged to generate a common edge value for the bunch graph.

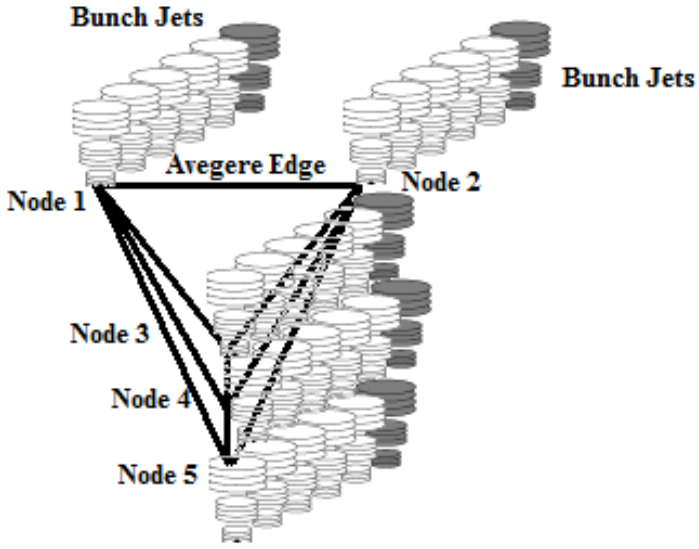

Figure 44 Face bunch graph generated using a set of model face image graph to represent an individual person

But for small database as taken for analysis, if the translation and rotation of candidate images are kept to minimum we can successfully use a minimum set of 7-8 image graphs of a person to generate a representative bunch graph.

4.1.4 Elastic Bunch Graph Matching

After a bunch graph is generated representative of each individual, each input face segment is used to generate a face graph which is then iteratively matched with each and every of the bunch graph by 'Elastic Bunch Graph Matching' to generate a quantitative similarity value between the input face graph and the model bunch graphs. (Wiskott, 1997)

Hence the initial set of bunch graphs has to be manually generated for each representative face/person. But after a model bunch graph is generated with 7-8 sample image graphs in our case to represent a person, further strength of the face bunch graph can be automatically updated as new face graph input will be matched with the existing model graphs and correlated to them.

4.1.4.1 Manual generation of graph

For the algorithm of facial recognition to recognize a facial image segment, it needs to match the image to a common set of marked facial images and find a similarity index between them.

This process is basically divided into two sets and the first part is manual generation of face bunch graph for a person with a set of model graphs. Hence, initially a training set of facial images is taken, where each image is marked to a corresponding person. This set of image is then used to manually generate a face bunch graph by three basic steps discussed below. (Kruger, 1997)

For our analysis, initial set of 7-8 Gabor wavelet transformed images are used as the initial training database, to generate an initial face bunch graph manually.

1. For a training set of face image segments extracted using face detection techniques discussed earlier, Gabor Kernel convolution is applied to generate a set of Gabor wavelet transformed images.
2. For this set of Gabor wavelet transformed images for a particular person, the fiducial points(left iris, right iris, nose, upper lip, chin) are manually marked

and the jet values of the Gabor wavelet transformed images of those points represents the node value of the fiducial points

3. The edges are drawn between each pair of fiducial nodes already marked and the magnitude of each edge is the length of the edge.
4. These set of nodes are stacked and the set of edges between similar fiducial points are averaged to generate the face bunch graph corresponding to a particular person.

This process is repeated for each individual who will form a model graph for recognition in the algorithm. In our analysis, for representative purpose we selected a set of 5 individuals and hence 5 face bunch graph is generated one for each of them.

In general the choice of fiducial points should be such that it evenly covers the whole face segment, but that requires a large set of fiducial points for each face which ultimately increases the complexity of analysis. But it is observed that for face finding the fiducial point at the outline of the face has more weight while for face recognition, fiducial points interior to the face region is important for identification procedure. Hence, as our algorithm here only concentrates for face recognition and face detection is already successfully completed, we choose out 5 fiducial points in the interior region of the face segment.

4.1.4.2 Matching using Graph Similarity Function

After the model bunch graph is generated for the representative set of faces, the goal is to match each individual bunch graph with an input image graph and rate it using a similarity function that will analyze the degree of similarity or distortion between the facial structures represented by the graphs.

The Elastic Bunch Graph Matching is responsible for evaluating the graph similarity between the input image graph and the face bunch graph of identical pose.

Thus for an image graph ***G*** with nodes ***n*** = 1, 2…N and edges ***e*** = 1, 2…, E matching is done between the corresponding parameters of the face bunch graph ***B*** as:

$$S_B(\mathcal{G}^I, \mathcal{B}) = \frac{1}{N}\sum_n \max_m \left(S_\phi(\mathcal{J}_n^I, \mathcal{J}_n^{Bm})\right) - \frac{\lambda}{E}\sum_e \frac{(\Delta\vec{x}_e^I - \Delta\vec{x}_e^B)^2}{(\Delta\vec{x}_e^B)^2},$$

Equation 20 Graph Similarity Measure between image graph and bunch graph

λ decides the relative importance between jets and metric structure, and is set at 1 for our analysis.

This process is Elastic Bunch Graph Matching because according to the pixel location of the fiducial points in the face bunch graph, the neighborhood of the same pixel in the image graph is elastically extended in order to find a set of fiducial node points for which the similarity maximizes between the image graph and the face bunch graph.

In our analysis, the locality about a fiducial point is taken to be of a range (+/- 5, +/-5) pixels in ***x*** and ***y*** directions. Hence, taking the nodal location of the jets in the face bunch graph and Base and considering the locality restrictions, the similarity estimation is done for 25 pixel point per node for the image graph, and the pixel point set where the similarity is maximum, is taken as the measure of similarity between the image graph and that particular face bunch graph.

4.1.5 Face Recognition based on Similarity Measure

After the algorithm designed to calculate the similarity measure between input image graph and the face bunch graph, the recognition is reduced to a simple neural network analysis.

As in case of our analysis, there is face bunch graph representative of 5 test people. Each of the face bunch graph is trained with a set of 4-5 image graphs belonging to the same person as the face bunch graph to find upper and lower threshold measure of similarity between them.

Hence, if an arbitrary input image graph is matched with the individual face bunch graph, then only if the similarity measure lies between the thresholds criteria set for that particular bunch graph using the training samples, then the image is of the same person of the face bunch graph. (Lades, 1993)

Our recognizer matches the input image with all the available face bunch graph and generates a similarity measure with each one of them (Sb1, Sb2…Sb5). Now this similarity measure value is input to the recognition thresholding limiter that generates binary output ‘1’ if the similarity measure is in the threshold limit and ‘0’ if out of threshold limit.

Now these outputs (O1, O2…O5) are multiplied by weights (W1, W2…W5) and then summed to generate a recognition index,

$$R=O1*W1+O2*W2+O3*W3+O4*W4+O5*W5$$

Equation 21 Recognition Index for the database used for analysis

Now if the Recognition index is 0, then the picture is not a match to the database and is unrecognized.

If the recognition index is same as any single weight (W1, W2,…W5), then it is perfect recognition and the image is recognized as one of the 5 person.

If the recognition index is sum of two or more weights, it is a case of over recognition and a manual definition needs to be done so that the thresholding limits are internally adjusted to remove over recognition.

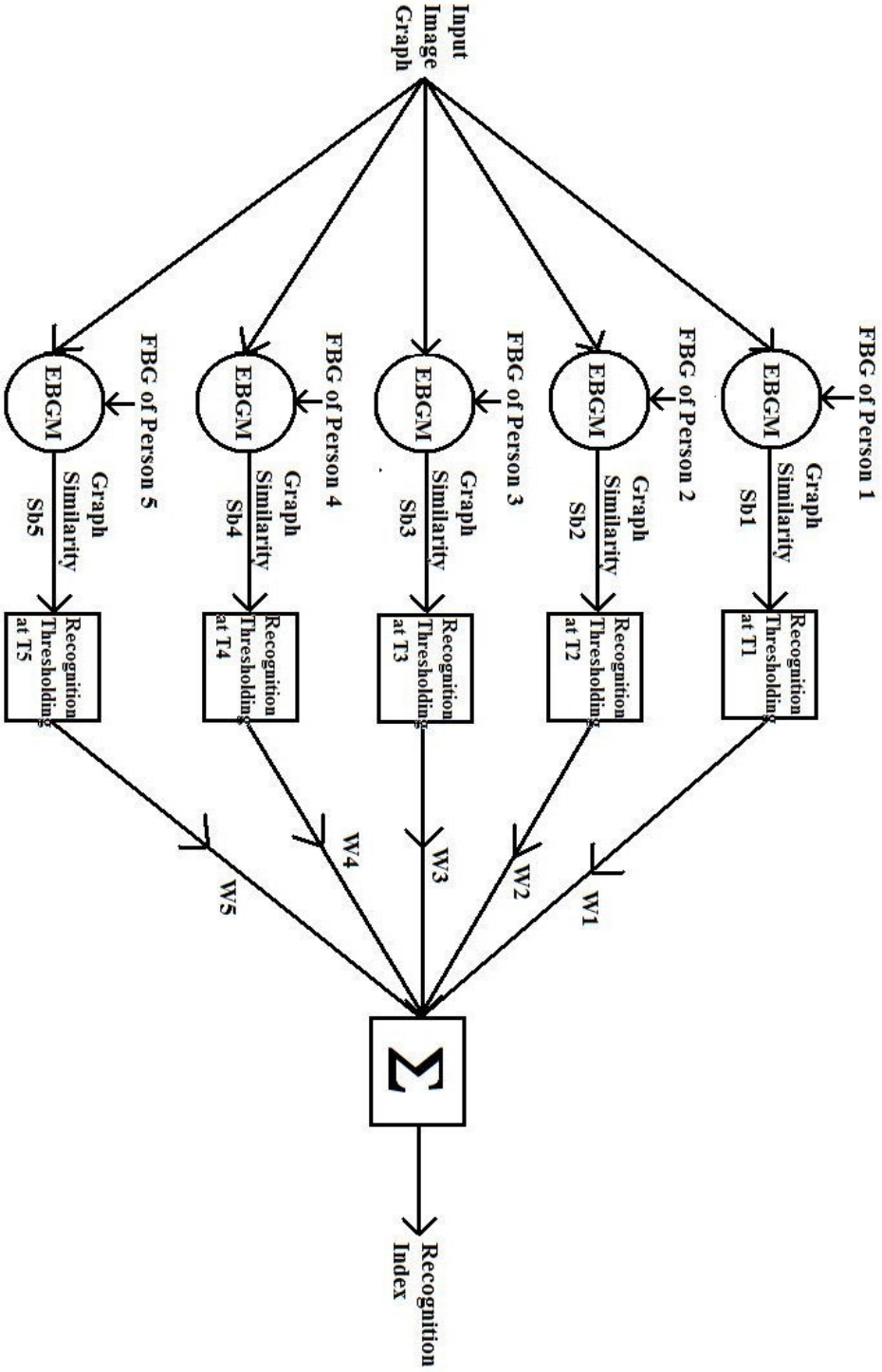

Figure 45 Recognition network

4.1.6 Result analysis for Facial Recognition

Figure 46 Input model image of Person 1 (left) and Person 2 (right)

Figure 47 Input model image of Person 3 (left) and Person 4 (right)

Figure 48 Input model image of Person 5

After the face bunch graph is generated for each of the 5 people aforementioned, they are trained with image graphs that are of same people to find a lower limit of threshold of the similarity measure, under which the image graph should be considered no match to the face bunch graph.

The thresholds are:

Person 1 (T1): 0.9986

Person 2(T2): 0.9989

Person 3(T3): 0.9983

Person 4(T4): 0.9989

Person 5(T5): 0.9990

Hence, when an arbitrary image graph is given as input to the recognizer, a similarity measure is calculated with all the 5 Face Bunch Graphs individually and if they match their individual threshold criteria, they are considered a match for

that face bunch graph, else not. Then it proceeds towards the calculation of a Recognition Index.

The examples of unmatched EBGM are:

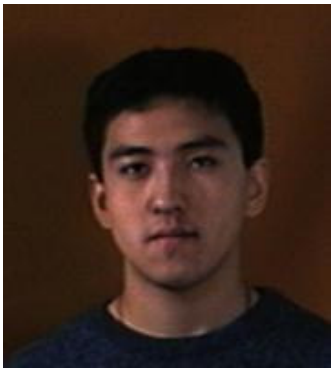

Figure 49 Person 2 face bunch graph and Person 4 input image graph have similarity measure of 0.9985<T2

Figure 50 Person 3 face bunch graph and Person 4 input image graph has similarity measure of 0.9968<T3

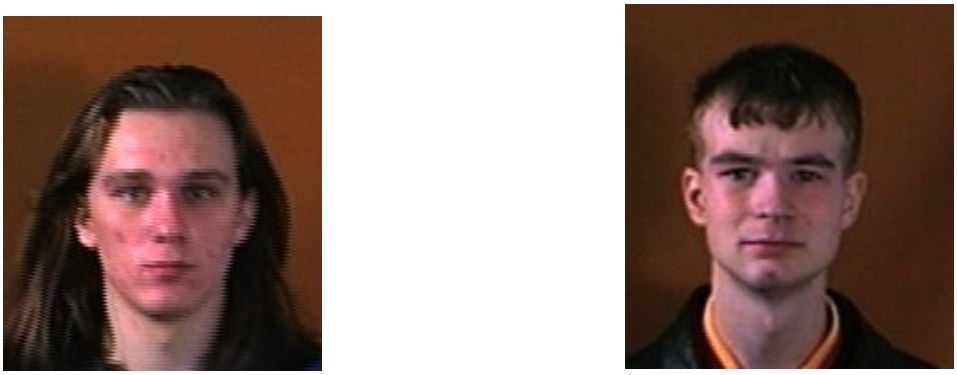

Figure 51 Person 3 face bunch graph and Person 4 input image graph has similarity measure of 0.9968<T3

Thus, the generated 'recognition index' has unique weight which successfully recognizes the person on the input image graph.

4.1.7 Matching Accuracy

Though the matching is successful for most of the images, it incorrectly identifies some images due to over identification.

This over identification reduces our accuracy to the range of 75% to 80% which can be increased only through high training data for finding the threshold of each EBGM condition and continuous user feedback update of the face bunch graph whenever an over recognition occurs.

Figure 52 Person 4 face bunch graph and Person 3 input image graph has similarity measure of 0.9991> T4, hence causes over identification of Person 3 as Person 3 and Person 4.

4.1.8 Limitations

As discussed earlier the EBGM algorithm implemented in our dataset is extremely prone to translation and rotation distortion and in our case the rotation of images is kept at 0 degrees and translation is restricted only in horizontal direction to less that 5 pixels so that the local feature does not moves out of the neighborhood.

4.2 Databases Used

- **The Sheffield face database (UMIST face database)** consisting of 564 images of 20 people, each covering a range of poses from profile to frontal views.

- **Face recognition data, University of Essex, UK (colour images)** which has data of 395 individuals (male, female) with 20 images per individual of people of various races between 18 to 20 years of age.

Chapter – V

Conclusion

The skin segmentation process followed by morphology is a fast process requiring very low computational time. However, the template matching requires more time but makes the overall efficiency very high. Out of 10 test images with a total of 80 faces, the face detector program stated above could recognize 76 faces with 4 faces being rejected because of size measures. Thus, the accuracy of such a system is under partially controlled environments with background different from skin-tone is very high, in this case around 0.95.

For face recognition system the Elastic Bunch Graph Matching algorithm is designed and implemented on a test database of approximately 30 facial image segments. The algorithm designed is robust to distortions in expressions and pose variations and hence can easily train the Face Bunch Graph representing an individual with a set of as low as 4-5 training images.

The recognition accuracy of EBGM is found to be in the range of 0.75 to 0.8 which is marginally better than neural network based recognition system, but requires far less training samples than the latter. But the matching is limited to low degree of translation and rotation distortions and the recognition accuracy falls rapidly for images with high degree of spatial distortions.

Chapter – VI

Bibliography

Beymer, D. (1994). Face recognition under varying pose. *Proc. IEEE Computer Vision and Pattern* , pp. 756-761.

Daugman, J. G. (1988). Complete discrete 2-D Gabor transform by neural networks for image analysis. *IEEE Trans. on Acoustics, Speech and Signal Processing* .

DeValois, R. L. (1988). *Spatial Vision.* Oxford Press.

Fleet, D. J. (1990). Computation of component image velocity from local phase information. *Int'l J. of Computer Vision* , 77-104.

Goldstein, A., Harmon, L., & Lesk, A. (May 1971). Identification of Human Faces. *IEEE Proceedings* , Vol. 59, No. 5, 748-760.

Gonzalez, R. C., & Woods, R. E. *Digital Image Processing.* New Jersey: Prentice Hall.

Jean-Marc Fellous, N. K. (1999). Face Recognition by Elastic Bunch Graph Matching. In L. Jain, *Intelligent Biometric Techniques in Fingerprint and Face Recognition* (pp. 355-396). CRC Press.

Kruger, N. P. (1997). Determination of face position and. *Image and Vision Computing* .

Lades, M. V. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. on Computers* , pp. 300-311.

Mohsin, W., Ahmed, N., & Mar, C.-T. (2003). *Face Detection Project.*

Movellan, J. *Introduction to Gabor Filters.*

NSTC Subcomittee on Biometrics. (n.d.). Retrieved from http://www.biometrics.gov/Documents/facerec.pdf

Petrou, M., & Kadyrov, A. *The Trace transform in a Nutshell.*

Pollen, D. A. (1981). *Phase relationship between adjacent simple cells in the visual cortex.* 1409-1411: Science.

Potzsch, M. K. (1997). Improving object recognition by transforming Gabor Filter responses. *Network: Computation in Neural Systems* , pp. 341-347.

Saveliev, P. (n.d.). *Euler Number - Computer Vision Primer*. Retrieved from Intelligent Perception: http://inperc.com/wiki/index.php?title=Euler_number

Sirovich, L., & Kirby, M. (1987). A Low-Dimensional procedure for Characterization of Human Faces. *Optical Soc. Am. A* , Vol. 4, No. 3, 519-524.

Support Vector Machine. (n.d.). Retrieved from Wikipedia: https://secure.wikimedia.org/wikipedia/en/wiki/Support_vector_machine

Theimer, W. M. (1994). Phase-based binocular vergence control and depth reconstruction using active vision. *CVGIP: Image Understanding* , 343-358.

Turk, M., & Pentland, A. (1991). Face Recognition Using Eigenfaces. *IEEE Proceedings* , 586-591.

Wiskott, L. F.-M. (1997). Face recognition by elastic bunch graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence* , pp. 775-779.

Xiong, Z. *An Introduction to Face Detection and Recognition.*